

⊕ public class Auto {

private int numeroRegistro;
private String modelo;
public double Precio { get; set; }

public Auto(int registro, String modelo) {
this.numeroRegistro = registro;
this.modelo = modelo;
}

public int NroRegistro() {
return this.numeroRegistro;
}

public override string ToString() {
return \$"{this.numeroRegistro}; {this.modelo}";
}

public void Importar(string linea) {

string[] splitResult = linea.Split(';');

this.numeroRegistro = Convert.ToInt32(splitResult[0].Trim());

this.modelo = splitResult[1].Trim();

this.Precio = Convert.ToDouble(splitResult[2].Trim());

}

+ public class Camion {

private DateTime fecha;

private int cantidad;

public int NroRegistro { get; set; }

public Camion(DateTime fecha, int cantidad) {

this.fecha = fecha;

this.cantidad = cantidad;

private Stack<Auto> transporte = new Stack<Auto>();

public void CargarVehiculo(Auto unAuto) {

if (transporte.Count < cantidad) {

transporte.Push(unAuto);

}

public Auto RetirarVehiculo() {

Auto v = null;

if (transporte.Count > 0) {

v = transporte.Pop();

return v;

public string[] VerCarga() {

string[] carga = new string[transporte.Count];

int i = 0;

foreach (Vehiculo v in transporte) {

carga[i++] = v.ToString();

return carga;

public double ValorAsegurado() {

return valorAsegurado;

public int CantidadVehiculos() {

return transporte.Count;

public override string ToString() {

return \$"{this.NroRegistro} - {this.fecha.ToString("dd-MM-yyyy")}

public double ValorAsegurado {

get { double valorAsegurado = 0;

foreach (Auto auto in transporte)

valorAsegurado += auto.Precio;

}

+) public Class Servicio {

```
private List<Auto> listaAutos = new List<Auto>();  
private List<Camion> listaCamiones = new List<Camion>();  
public int NroOrden { get; set; }
```

```
public int GenerarCamion(DateTime fecha, int capacidad) {
```

```
    Camion camion = new Camion(fecha, capacidad);  
    listaCamiones.Add(camion);  
    NroOrden++;  
    camion.NroRegistro = NroOrden;
```

```
    return camion.NroRegistro;  
}
```

```
public void RecibirCamion(Camion unCamion) {
```

```
    listaCamiones.Add(unCamion);  
}
```

```
public Auto DescargarCamion(int nroOrden) {
```

```
    Camion camionBuscado = BuscarCamion(nroOrden);
```

```
    Auto v = null;
```

```
    if (camionBuscado != null) {
```

```
        v = camionBuscado.RetirarVehiculo();
```

```
        listaAutos.Add(v);
```

```
    }  
    return v;
```

```
private Camion BuscarCamion(int nro) {
```

```
    int i = 0;
```

```
    int pos = -1;
```

```
    while (pos == -1 && i < listaCamiones.Count) {
```

```
        if (listaCamiones[i].NroRegistro == nro) {
```

```
            pos = i;
```

```
            i++;
```

```
        }  
        return pos;
```

```
    }  
    Camion c = null;
```



```

if (pos != -1) {
    C = listaCamiones[pos];
}
return C;
}

```

```

public void CargarCamion (int nroOrden, Auto unAuto) {
    Camion camionBuscado = BuscarCamion(nroOrden);
    if (camionBuscado != null) {
        camionBuscado.CargarVehiculo(unAuto);
        listaAutos.Remove(unAuto);
    }
}

```

```

public void CerrarCamion (int nro) {
    Camion camion = BuscarCamion(nro);
    if (camion != null) {
        string path = camion.ToString() + ".csv";
        FileStream fs = null;
        StreamWriter sw = null;
        try {
            fs = new FileStream(path, FileMode.Create, FileAccess.Write);
            sw = new StreamWriter(fs);
            sw.WriteLine("nroRegistro; modelo");
            foreach (string linea in camion.VerCarga()) {
                sw.WriteLine(linea);
            }
        } catch (Exception ex) { throw new ex; }
        finally {
            if (sw != null) sw.Close();
            if (fs != null) fs.Close();
        }
    }
}

```


④ public class FormPrincipal {

Servicio servicio = new Servicio();

int num;

private void btnCrearCamion_Click() {

DateTime fecha = dtbFecha.Value;

int capacidad = Convert.ToInt32(tbCapacidad.Text);

int num = servicio.GenerarCamion(fecha, capacidad);

MessageBox.Show("Camión Creado");

#region Limpiar campos

tbFecha.Clear();

dtb.Value = DateTime.Now;

#endregion

}

private void btnCargar_Click() {

Auto vehiculoSelected = lsAutos.SelectedItem as Auto;

if (vehiculoSelected != null) {

servicio.CargarCamion(num, vehiculoSelected);

lsAutos.Items.Remove(vehiculoSelected);

MessageBox.Show("Vehículo cargado");

} else {

MessageBox.Show("Debes seleccionar un vehículo de la lista para cargarlo");

}

private void btnDescargar_Click() {

Auto vehiculo = servicio.DescargarCamion(num);

lsAutos.Items.Add(vehiculo);

}


```

private void btnCerrarCamion - Click(-) {
    servicio.CerrarCamion(num);
}

```

```

private void btnRecibirCamion - Click(-) {
    openFileDialog.Filter = "(csv) | *.csv | (txt) | *.txt";
    openFileDialog.Title = "IMPORTACION DE VEHICULOS";
    if (openFileDialog.ShowDialog() == DialogResult.OK) {
        FileStream fs = null;
        StreamReader sr = null;
        string path = openFileDialog.FileName;
        try {
            fs = new FileStream(path, FileMode.Open, FileAccess.Read);
            sr = new StreamReader(fs);
            sr.ReadLine();
            while (!sr.EndOfStream) {
                string lineas = sr.ReadLine();
                string[] splitResult = lineas.Split(';');
                int nroRegistro = Convert.ToInt32(splitResult[0].Trim());
                string modelo = splitResult[1].Trim();
                double precio = Convert.ToDouble(splitResult[2].Trim());

                Auto auto = new Auto(nroRegistro, modelo);
                auto.Precio = precio;

                lstRecibidos.Items.Add(auto);
            }

            Camion camionRecibido = new Camion(DateTime.Now, lstRecibidos.Items.Count);
            sistema.RecibirCamion(camionRecibido);
            foreach (Auto auto in lstRecibidos.Items) {
                servicio.CargarCamion(num, auto);
            }
        } catch (Exception ex) {
            MessageBox.Show(ex.Message, "ERROR EN LA IMPORTACION",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        } finally {
            if (sr != null) sr.Close();
            if (fs != null) fs.Close();
        }
    }
}

```



```

private void btnDescargarCamion_Click(-) {
    IF (num > 0) {
        Auto auto = servicio.DescargarCamion(num);
        while (auto != null) {
            lsAutos.Items.Add(auto);
            lsRetirados.Items.Remove(auto);
            auto = servicio.DescargarCamion(num);
        }
    }
}

```

```

private void FormPrincipal_Load(-) {
    ⊕ Deserializo
    List<Auto> autos = new List<Auto> {

```

```

        new Auto(100, "Audi AA"),
        new Auto(101, "Audi BB"),
        new Auto(102, "Corso B2"),
        new Auto(103, "Corso Full")
    };

```

```

    Camion camion = new Camion(DateTime.Now, 6);
    camion.NroRegistro = sistema.NroOrden;
    servicio.RecibirCamion(camion);

```

```

    foreach (Auto auto in autos) {
        servicio.CargarCamion(
            camion.NroRegistro,
            auto);
    }

```

```

    lsAutos.Items.AddRange(autos.ToArray());

```

```

⊕ FileStream fs = null;
try {

```

```

    fs = new FileStream("datos.bin", FileMode.Open, FileAccess.Read);
    BinaryFormatter bf = new BinaryFormatter();
    servicio = bf.Deserialize(fs) as Sistema;

```

```

} catch (Exception ex) {
    MessageBox.Show(ex.Message, "ERROR AL DESERIALIZAR",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
} finally {
    if (fs != null) fs.Close();
}

```

Funcionamiento
Lógico


```
private void FormPrincipal_Closing(...) {
```

```
    FileStream FS = null;
```

```
    try {
```

```
        FS = new FileStream("datos.bin", FileMode.Create, FileAccess.Write);  
        BinaryFormatter bf = new BinaryFormatter();  
        bf.Serialize(FS, serwin);
```

```
    } catch (Exception ex) {
```

```
        MessageBox.Show(ex.Message, "ERROR AL SERIALIZAR", MessageBoxButtons.Ok,  
                          MessageBoxIcon.Error);
```

```
    } finally
```

```
    {  
        if (FS != null) FS.Close();  
    }
```

```
}
```

```
}
```